

---

# 다양한 모듈 활용

9주차\_02

한 동 대 학 교  
김 경 미 교수

# 그래픽 처리 Tkinter module

---

- 웹 프로그래밍을 할 수 있도록 GUI를 제공
- Tool Kit for interactive programming
- ttk library를 같이 활용
- Tkinter widgets
  - Button
  - Checkbutton
  - Entry
  - Frame
  - Label
  - Menubutton

# Checkboxes

```
from tkinter import *
```

```
top = Tk()
```

```
CheckVar1 = IntVar()
```

```
CheckVar2 = IntVar()
```

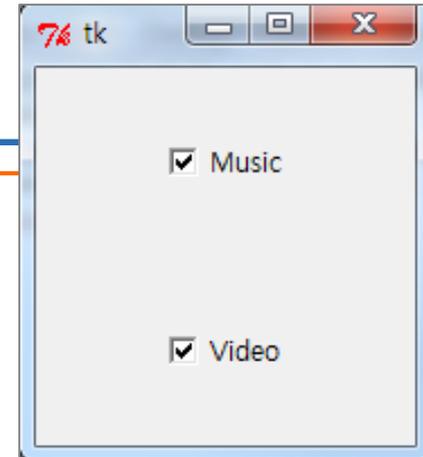
```
C1 = Checkbutton(top, text = "Music", variable = CheckVar1, onvalue = 1,  
                 offvalue = 0, height=5, width = 20)
```

```
C2 = Checkbutton(top, text = "Video", variable = CheckVar2, onvalue = 1,  
                 offvalue = 0, height=5, width = 20)
```

```
C1.pack()
```

```
C2.pack()
```

```
top.mainloop()
```



# Entry widget, 입력 받기

```
from tkinter import *
```

```
top = Tk()
```

```
L1 = Label(top, text="User Name")
```

```
L1.pack(side = LEFT)
```

```
E1 = Entry(top, bd =5)
```

```
E1.pack(side = RIGHT)
```

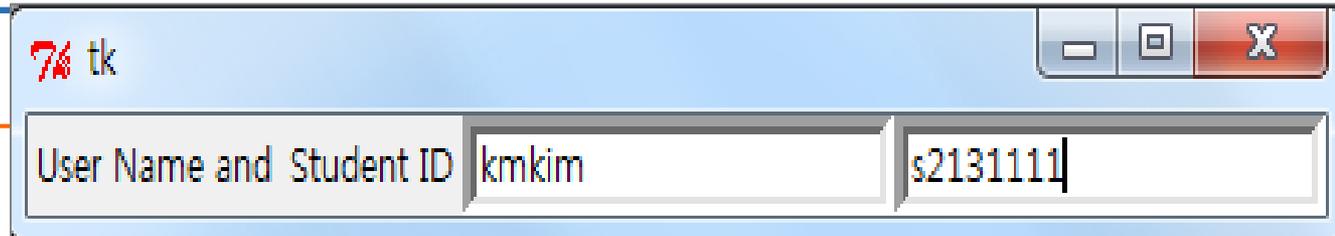
```
L2 = Label(top, text="Student ID")
```

```
L2.pack(side = LEFT)
```

```
E2 = Entry(top, bd =5)
```

```
E2.pack(side = RIGHT)
```

```
top.mainloop()
```



# Displaying Images

```
from tkinter import *  
from tkinter import ttk  
  
frame = ttk.Frame()  
frame['padding'] = (5,10)  
frame['borderwidth'] = 2  
frame['relief']='sunken'  
frame.grid()  
  
label = ttk.Label(text='Full name;')  
  
image = PhotoImage(file='cat_dog.gif')  
label['image'] = image  
label.grid()
```



# 다양한 모듈

---

다양한 모듈	
sys	프로그램 실행 환경과 관련한 정보를 제공
os	파일, 프로세스, 디렉토리 등 다양한 운영체제 기능 제공
htmllib	HTML 분석 모듈
cgi	cgi-bin에서 파이썬으로 웹 응용 작성을 도움

# Operating System Modules (1)

---

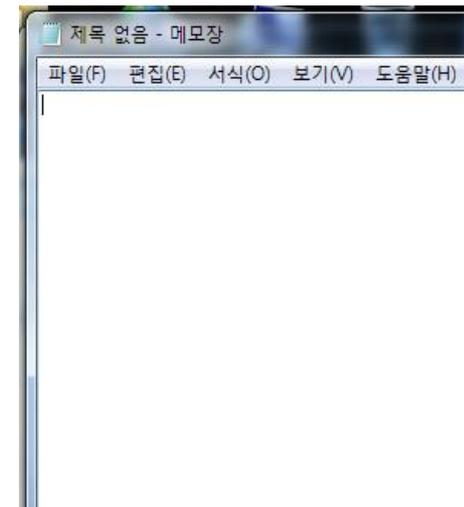
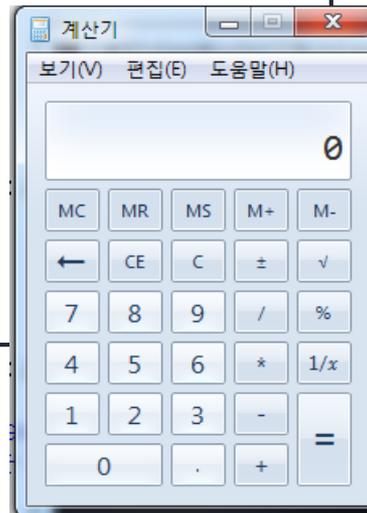
- **os module**

- 다수의 운영체제 함수들에 통일된 인터페이스를 제공

```
>>> from os import *
```

```
>>> system('calc')
```

```
>>> system('notepad')
```



# Operating System Modules (2)

# Using the os.path module to handle filename

```
import os
filename = "my/little/pony"
```

```
print("using", os.name, "...")
```

```
print("split", "=>", os.path.split(filename))
```

```
print("splittext", "=>", os.path.splittext(filename))
```

```
print("dirname", "=>", os.path.dirname(filename))
```

```
print("basename", "=>", os.path.basename(filename))
```

```
print("join", "=>", os.path.join(os.path.dirname(filename), os.path.basename(filename)))
```

```
>>>
===== RESTART: E:/1_Works/2017Work/KMooC강의
=====
using nt ...
split => ('my/little', 'pony')
splittext => ('my/little/pony', '')
dirname => my/little
basename => pony
join => my/little#pony
>>>
```

# 사용자가 만드는 모듈 1

```
# filename; fibo.py
# Fibonacci numbers module

def fib(n) :
    if n == 0 :
        return 0
    elif n == 1 :
        return 1
    else :
        return fib(n-1) + fib(n-2)

def ifib(n) :
    a = 0
    b = 1
    for i in range(n) :
        a = b
        b = a + b
    return a
```

Fibo.py안에 함수 2개를 선언한다  
이 함수들을 사용하고 싶은 곳에서,  
`import fibo`  
쓰면, 저장되어 있는 함수 2개 사용 가능하다

```
>>> import fibo
>>> f=fibo
>>> f.fib(11)
89
>>> f.ifib(11)
1024
>>> |
```

# 사용자가 만드는 모듈 2

---

- 자주 사용하는 함수들을 모아서 모듈로 사용 가능
- 관련 있는 함수들은 모아서, 몇 개의 사용자 정의 모듈을 만들면
  - 코딩 소요 시간 감소
  - 다른 사람과 같이 작업할 때, 공유해야 함

# 연습문제 1

---

- 자주 사용하는 함수 3개를 저장하여 'freq.py'에 저장한다
- `import freq` 사용하여 저장된 함수 3개를 사용해 본다

# 연습문제 1 코드

---

```
from datetime import date

def cal_birthday(month, day):
    today = date.today()
    birthday = date(today.year, month, day)
    due = birthday - today
    if due.days < 0 :
        next_birthday = date(today.year + 1, birthday.month, birthday.day)
        due = next_birthday - today

    print("생일까지 남은 날짜는: ", due.days)

import math
import cmath

def deter(a, b, c):
    return math.pow(b, 2) - 4*a*c

# continue to..
```

# 연습문제 1 코드

```
def roots_formula(a, b, c):
    if deter(a,b,c) >= 0:
        root01 = (-b + math.sqrt(deter(a,b,c)))/ (2*a)
        root02 = (-b - math.sqrt(deter(a,b,c)))/ (2*a)
    else:
        root01_real = -b/(2*a)
        root01_imag = (math.sqrt(math.fabs(deter(a,b,c))))/ (2*a)
        root02_real = -b/(2*a)
        root02_imag = (math.sqrt(math.fabs(deter(a,b,c))))/ (2*a)
        root01 = root01_real + root01_imag * 1j
        root02 = root02_real - root02_imag * 1j

    return [root01, root02]
```

```
>>> import freq
>>> f=freq
>>> print(f.roots_formula(1,2,3))
[(-1+1.4142135623730951j), (-1-1.4142135623730951j)]
>>> f.cal_birthday(11, 24)
생일까지 남은 날짜는: 147
>>>
```

# 다양한 모듈들 1

---

String 관련 Module	
Re	String을 효과적으로 분석, 처리하는 정규식 지원
struct	C언어 API 지원, struct 다루고 binary file 처리 지원
Difflib	file비교 관련 module
text wrap	word-wrapping들 텍스트 처리 지원
codecs	텍스트 인코딩 관련 module

# 다양한 모듈들 2

---

Data type 관련 Module	
calendar	달력, 윤달 확인, 주 단위 시작 종료 추출
bisect	stack, queue 관련 지원
array	이미지나 음성 파일 처리 시 사용 배열 지원
copy	복잡한 오브젝트의 복사본 관리
pprint	내포한 리스트나 딕셔너리 보기 쉽게 출력
sets	임의의 집합 관리

# 다양한 모듈들 3

---

인터넷 데이터 처리 Module	
mimify	메일 메시지의 인코딩 및 디코딩
Binascii	binary data와 ASCII 간의 변환
Binhex	MAC용 binhex 의 압축 및 압축해제
quopri	복잡한 오브젝트의 복사본 관리

# 다양한 모듈들 4

---

마크업 처리 Module	
htmllib	HTML 분석 module
sgmlib	SGML 분석 module
xml.sax, xml.dom	XML 분석 module
formatter	HTML, XML 및 기타 형식으로 출력 지원

# 다양한 모듈들 5

---

## 인터넷 프로토콜 처리 Module

cgi	cgi-bin에서 파이썬으로 웹응용 작성 지원
urllib, urlparse	URL 열고 결과를 파싱 지원
httplib, ftplib, gopherlib	HTTP, FTP 프로토콜 client 사용 지원
Poplib, imaplib	메일 읽는 POP3, IMAP 지원
SocketServer	TCP, UDP server 제작 지원
SimpleHTTPServer, CGIHTTPServer, BasedHTTPServer	간단한 웹 서버 구축 지원
smtplib	메일 전송을 위한 SMTP/ESMTP client 구현

# 숙제

---

- 연습문제 1을 입력한 코드와
- 실행 결과를 캡처하여 게시판에 올리시오

# 요약

---

- tkinter 모듈 활용한다
- os 모듈 활용한다
- 사용자가 만드는 모듈을 정의한다

---

# 감사합니다

9주차\_02 다양한 모듈 활용